

TestCase-rules

Лучшие практики по оформлению тест-кейсов, требования к оформлению тестов и основные атрибуты.

Тест-кейс

— набор входных значений, предусловий выполнения, ожидаемых результатов и постусловий выполнения, разработанный для определённой цели или тестового условия, таких как выполнения определённого пути программы или же для проверки соответствия определённому требованию.

Правила написания тест-кейсов

1. Заголовок:

- должен быть чётким, кратким, понятным и однозначно характеризующим суть тест-кейса;
- не может содержать выполняемые шаги и ожидаемый результат.

2. Предусловие:

- может содержать полную информацию о состоянии системы или объекта, необходимом для начала выполнения шагов тест-кейса;
- может содержать ссылки на информационные источники, которые необходимо изучить перед прохождением тест-кейса (инструкции, описание систем...);
- не может содержать ссылки на тестируемый ресурс, если у информационной системы более одной среды (прод, тест, препрод...), данная информация должна быть вынесена в инструкцию, и ссылка приложена в предусловии;
- не может содержать данные для авторизации, данная информация должна быть вынесена в инструкцию, и ссылка приложена в предусловии;
- не может содержать выполняемые шаги и ожидаемый результат, если нам нужно, чтобы до выполнения шагов проверки у нас была открыта

главная страница, то мы в предусловии указываем «открыта главная страница сайта»;

- не может содержать ожидаемый результат.

3. Шаги проверки:

- должны быть чёткими, понятными и последовательными;
- следует избегать излишней детализации шагов. Правильно: «ввести в поле число 12». Неправильно: «нажать на клавиатуре на цифру '1', следующим шагом нажать на клавиатуре на цифру '2'»;
- должны использоваться безличные глаголы. Правильно: нажать, ввести, перейти. Неправильно: нажмите, введите, идите;
- не должно быть комментариев и пояснений, если есть необходимость привести мини-инструкцию, то оформляем инструкции в базе-знаний и в предусловии ссылаемся на неё;
- не должно быть жёстко прописанных статических данных (логины, пароли, имена файлов) и примеров, для исключения эффекта пестицида.

4. Ожидаемый результат:

- должен быть у каждого шага проверки;
- должно быть кратко и понятно описано состояние системы или объекта, наступающее после выполнения соответствующего шага;
- не должно быть избыточного описания.

5. Общие требования к тест-кейсам:

- язык описания тест-кейсов должен быть понятен широкому кругу пользователей, а не узкой группе лиц;
- тест-кейс должен быть максимально независим от других тест-кейсов и не ссылаться на другие тест-кейсы (лучшая практика, когда зависимостей нет вообще);
- тест-кейсы группируются в функциональные блоки по их назначению;
- в тест-кейсах проверяющих работу функционала скриншотов быть не должно, иначе вы будете посвящать сотни часов на изменение всех скриншотов в тысячах тест-кейсах при изменении интерфейса

тестируемой программы. Скриншоты могут быть добавлены только в тест-кейсы проверяющие отображение страниц и форм.

На самом деле правила простые, однако их не так-то просто соблюдать. Если же придерживаться данных правил, то тест-кейсы будут легко поддерживаемыми, легко читаемыми, не будут вызывать отторжения и могут быть использованы всеми участниками команды в процессе разработки программного обеспечения.

Лучшие практики в написании тест-кейсов

- Перед созданием нового тест-кейса убедитесь, что он не дублирует ни один из уже существующих в системе.
- Убедитесь, что тест-кейс покрывает 100% требований, которые вы должны проверить.
- Помните о теории методов тестирования, таких как анализ граничных значений, деление эквивалентности, техника перехода состояния, угадывание ошибок.
- Помимо требований к системе, всегда помните о конечном пользователе, который будет взаимодействовать с системой.
- Не забудьте указать учетные данные, если они необходимы для выполнения теста.
- Позаботьтесь о тестировщиках, которые будут работать с этим тест-кейсом в будущем. В частности, убедитесь, что все ссылки верны и кликабельны. Пожалуйста, не используйте ссылки на продакшен.
- Используйте повелительное наклонение, например: *«Перейдите на главную страницу»*, *«Введите данные»*, *«Кликните»* и т. д. Такая формулировка упрощает понимание этапов тестирования и ускоряет их выполнение.

Атрибуты тест-кейса для ручного тестирования

Как и все тестировочные документы, тест-кейс имеет определенный формат. Он содержит следующие атрибуты:

1. **ID** — уникальное сочетание букв и цифр.
2. **Заголовок** — основная идея тест-кейса, краткое описание его сути.
Например, заголовок тест-кейса для ручного тестирования страницы входа может выглядеть следующим образом: «*Проверить вход пользователя с корректными данными*».
3. **Предусловия** — список действий, которые необходимо выполнить перед выполнением тест-кейса. При необходимости здесь могут указываться учетные данные.
4. **Шаги** — описание действий, необходимых для проверки.
5. **Постусловия** — список действий, возвращающих систему в исходное состояние (указывается при необходимости).
6. **Ожидаемый результат** — то, что мы ожидаем получить после успешного выполнения тест-кейса.
7. **Фактический результат** — то, что мы получаем после выполнения тест-кейса (указывается при необходимости).
8. **Статус** — Success (успех), Failed (провал), Blocked (блокировка) (указывается при необходимости).

Кроме того, для некоторых тест-кейсов могут потребоваться дополнительные атрибуты:

- **Требования к среде** — специальное оборудование, программное обеспечение и т. п. вещи, необходимые для выполнения тест-кейса и не перечисленные в соответствующей спецификации проекта тестирования.
- **Специальные процедурные требования** — особые процедуры настройки, выполнения или очистки, уникальные для этого тест-кейса.
- **Межкейсовые зависимости** — тест-кейсы, которые нужно выполнить перед этим тест-кейсом.

Позитивные, негативные и деструктивные тест-кейсы

- **В позитивных тест-кейсах** используются корректные входные данные и сценарии ожидаемой работы системы. Цель здесь — убедиться, что программный продукт выполняет то, что должен делать, и что система не выдаст ошибку, если это не предусмотрено.
В целом позитивное тестирование гарантирует, что система соответствует требованиям при позитивных сценариях нормального использования. Например, если поле пароля принимает десять символов, пользователь должен иметь возможность создать такой пароль.
- **Негативные тест-кейсы** используют некорректные входные данные и проверяют, не делает ли программа того, чего не должна делать. Негативное тестирование призвано гарантировать, что при получении некорректных входных данных система не будет работать по нормальному сценарию (например, выбросит ошибку).
Если вернуться к нашему примеру, пользователь не должен иметь возможность создать пароль, состоящий из 11 символов.
- **Деструктивные тест-кейсы** создаются, чтобы узнать предел прочности системы. Нагрузочное тестирование — распространенный вариант деструктивного тестирования.
Для деструктивного тестирования QA-специалисты могут применять следующие методы:
 - создание большой нагрузки на систему, чтобы это привело к ее отказу;
 - злонамеренное внедрение JavaScript в веб-форму;
 - фаст-кликер в попытках взломать веб-страницу и т. д.